

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	6	return pointer\$1 breakpoint\$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	WITH	OFF	2005/01/27 11:11
L2	45	return pointer\$1 breakpoint\$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	SAME	OFF	2005/01/27 11:11
L7	24	(US-5142679-\$ or US-5313616-\$ or US-5333304-\$ or US-5491808-\$ or US-5504901-\$ or US-5509131-\$ or US-5581697-\$ or US-5687375-\$ or US-5838976-\$ or US-5903758-\$ or US-5950003-\$ or US-6026235-\$ or US-6026236-\$ or US-6126328-\$ or US-6161196-\$ or US-6189141-\$ or US-6223339-\$ or US-6263488-\$ or US-6327704-\$ or US-6334213-\$ or US-6629123-\$ or US-6721941-\$ or US-6760907-\$).did. or (US-20030037318-\$).did.	USPAT; DERWENT	OR	OFF	2005/01/27 11:40
L8	10	7 and breakpoint\$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/01/27 11:41
L9	186	717/129.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/01/27 11:47
L10	271	717/130.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/01/27 11:48
L11	279	717/131.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/01/27 11:48
L12	50	717/133.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/01/27 11:50

L13	220	714/34.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/01/27 11:50
-----	-----	--------------	---	----	-----	------------------


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

+(function +exits) +and +breakpoints +and +(return +pointers)

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **function exits** and **breakpoints** and **return pointers**

Found 59 of 148,786

Sort results by

relevance

Display results

expanded form

☒ [Save results to a Binder](#)
☒ [Search Tips](#)
☐ [Open results in a new window](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 20 of 59

 Result page: [1](#) [2](#) [3](#) [next](#)

 Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [A structural view of the Cedar programming environment](#)

Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann

 August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 8 Issue 4

 Full text available: [pdf\(6.32 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

### 2 [The benefits and costs of DyC's run-time optimizations](#)

Brian Grant, Markus Mock, Matthai Philipose, Craig Chambers, Susan J. Eggers

 September 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 22 Issue 5

 Full text available: [pdf\(1.59 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

DyC selectively dynamically compiles programs during their execution, utilizing the run-time-computed values of variables and data structures to apply optimizations that are based on partial evaluation. The dynamic optimizations are preplanned at static compile time in order to reduce their run-time cost; we call this staging. DyC's staged optimizations include (1) an advanced binding-time analysis that supports polyvariant specialization (enabling both single-way and multi ...

**Keywords:** dynamic compilation, specialization

### 3 [Debugging optimized code without being misled](#)

Max Copperman

 May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 16 Issue 3

 Full text available: [pdf\(2.57 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Correct optimization can change the behavior of an incorrect program; therefore at times it is necessary to debug optimized code. However, optimizing compilers produce code that impedes source-level debugging. Optimization can cause an inconsistency between where the user expects a breakpoint to be located and the breakpoint's actual location. This article describes a mapping between statements and breakpoint locations that ameliorates this problem. The mapping enables debugger b ...